

Partie Persistance :

Nous avons utilisé la persistance avec un fichier binaire dans notre projet. Cela nous permet d'enregistrer et stocker les données en permanence dans un fichier binaire (BinaryFormatter) pour pouvoir les récupérer au prochain lancement de l'application.

Pour pouvoir utiliser la persistance, nous avons créé une interface IPersistanceManager qui contient deux méthodes : 'BD_Utilisateur ChargDonnées()' et 'void SauvegardeDonnées(BD_Utilisateur bdUser)'

Les méthodes ne sont pas définies dans l'interface, mais dans une autre classe nommée SauvegardeChargement qui implémente la classe IPersistanceManager

Cette classe SauvegardeChargement permet de définir ce que font les deux méthodes

La méthode BD_Utilisateur ChargDonnées() permet de charger les données qui sont dans le fichier binaire pour pouvoir récupérer les données qui sont à l'intérieur au lancement de l'application

La méthode void SauvegardeDonnées(BD_Utilisateur bdUser) permet de sauvegarder les données dans le fichier binaire.

Pour pouvoir utiliser la persistance dans notre projet, on a implémenté l'interface IPersistanceManager dans la classe Manager et redéfini les deux méthodes de l'interface pour les faire correspondre avec les classes de notre projet

Pour créer les données qui seront utilisées dans l'application, il faut implémenter l'interface IPersistanceManager dans la classe Stub et redéfinir les deux méthodes de l'interface.

La méthode de chargement des données dans le Stub permet de créer les collections de données par défaut qui seront utilisées pour tester l'application.

La méthode de sauvegarde des données dans le Stub permet d'afficher un message de Debug quand une sauvegarde est demandée.

Pour pouvoir charger les données du stub dans le fichier, il faut instancier le manager avec BD_Utilisateur créer dans le stub (dans le fichier App.xaml.cs)

Ensuite, une fois que le fichier est rempli, nous pouvons lancer le projet avec le contracteur par défaut de manager qui permet de lire les données du fichier

A chaque modification ou ajout, l'application fait appelle a la sauvegarde

Pour pouvoir sauvegarder des classe dans un fichier, il faut rappeler à cette méthode quelle est Serializable grâce à la ligne ajoutée "[Serializable()]" dans les classes BD_Utilisateur, Utilisateur et Session.